

**In the Claims:**

Please amend claims 17, 18, 25, 31, 36-38 and 41-43 as indicated below.

1. – 16. (Canceled)

17. (Currently amended) A computer system, comprising:

a processor; and

a memory, wherein the memory comprises program instructions executable on the processor to implement a Document Exchange (XDOC) framework for processing in-bound and out-bound documents in an electronic procurement system, wherein the XDOC framework comprises:

an Extensible Markup Language (XML) content configuration file module ~~configured to~~ that provides XML content gathered from a plurality of in-bound documents and applied to out-bound documents responsive to said in-bound documents;

a conduit file module, coupled to said XML content configuration file module, ~~configured to~~ that receives files in a first XML format and generate corresponding files in a second XML format; and

a persistent object framer (POF) module, coupled to said XML content configuration file module, ~~configured to~~ that maintains data persistence of files stored in a database external to said XDOC framework in said electronic procurement system, wherein said files stored in said database correspond to said XML content in said in-bound documents and said out-bound documents respectively.

18. (Currently amended) The computer system of Claim 17, wherein said XML content configuration file module further comprises XML content retrieving logic ~~configured to~~ that retrieves XML content from data sources that are external and internal to said XML content configuration file module.

19. (Previously presented) The computer system of Claim 17, wherein said in-bound documents are purchase orders describing information related to buyer requests in said electronic procurement system, and where said out-bound documents are purchase orders describing supplier goods and services in said electronic procurement system.

20. (Previously presented) The computer system of Claim 19, wherein said first XML file format is an Open Buying on the Internet (OBI) Standards compliant XML file.

21. (Previously presented) The computer system of Claim 17, wherein said files stored in said database are defined as data objects with related attributes.

22. (Previously presented) The computer system of Claim 21, wherein said attributes define in granular details, the contents of the data objects stored in said database.

23. (Previously presented) The computer system of Claim 20, wherein said markup content is substantially compliant with Handheld Device Markup Language content.

24. (Previously presented) The computer system of Claim 20, wherein said markup content comprises a content suitably adapted to interact with an Internet browser of a computer system.

25. (Currently amended) The computer system of Claim 24, wherein said XML content configuration file module[[s]] further comprises availability logic ~~configured to~~

that determines whether content selected by a user in said in-bound documents is available for presentation in said out-bound documents.

26. – 30. (Canceled)

31. (Currently amended) A method, comprising:

one or more computers performing:

receiving one or more input documents in a markup language, wherein the input documents comprise a purchasing request, wherein each input document comprises one or more tag names specifying purchasing parameters for the purchasing request;

parsing each of the one or more input document to identify each of the one or more tag names;

generating one or more data objects each corresponding to a respective tag name;

identifying one or more purchasing parameters related to one or more of the identified tag names; and

generating one or more output documents in the markup language specifying a purchasing order to a supplier corresponding to the purchasing request specified in the one or more input documents, wherein the one or more output documents comprise:

one or more tag names corresponding to the one or more identified purchasing parameters, and

one or more data attributes each corresponding to one of the one or more tag names corresponding to the one or more identified purchasing parameters.

32. (Previously presented) The method of claim 31, further comprising:

generating a data object tree of the one or more data objects based on relationships between the corresponding tag names as defined in the one or more input documents; and

wherein said identifying one or more purchasing parameters comprises traversing the data object tree comparing each data object of the tree to one or more data objects in a database specifying the one or more purchasing parameters.

33. (Previously presented) The method of claim 32, wherein said identifying one or more purchasing parameters further comprises identifying the one or more purchasing parameters based upon a configuration file defining relationships between the purchasing parameters and the one or more tag names.

34. (Previously presented) The method of claim 33, wherein the configuration file is one of a plurality of configuration files, wherein each configuration file defines different relationships between the purchasing parameters and the one or more tag names.

35. (Previously presented) The method of claim 31, wherein the markup language is eXtensible Markup Language (XML).

36. (Currently amended) A device, comprising:

a processor; and

a memory coupled to the processor, wherein the memory comprises program instructions ~~configured~~ that when executed by the processor execute to:

receive one or more input documents in a markup language, wherein the input documents comprise a purchasing request, wherein each input document comprises one or more tag names specifying purchasing parameters for the purchasing request;

parse each of the one or more input document to identify each of the one or more tag names;

generate one or more data objects each corresponding to a respective tag name;

identify one or more purchasing parameters related to one or more of the identified tag names; and

generate one or more output documents in the markup language specifying a purchasing order to a supplier corresponding to the purchasing request specified in the one or more input documents, wherein the one or more output documents comprise:

one or more tag names corresponding to the one or more identified purchasing parameters, and

one or more data attributes each corresponding to one of the one or more tag names corresponding to the one or more identified purchasing parameters.

37. (Currently amended) The device of claim 36, wherein the program instructions, when executed, ~~are further configured~~ execute to:

generate a data object tree of the one or more data objects based on relationships between the corresponding tag names as defined in the one or more input documents; and

wherein in said identifying one or more purchasing parameters the program instructions ~~are further configured~~ execute to traverse the data object tree comparing each data object of the tree to one or more data objects in a database specifying the one or more purchasing parameters.

38. (Currently amended) The device of claim 37, wherein in said identifying one or more purchasing parameters the program instructions ~~are further configured~~ execute to identify the one or more purchasing parameters based upon a configuration file defining relationships between the purchasing parameters and the one or more tag names.

39. (Previously presented) The device of claim 38, wherein the configuration file is one of a plurality of configuration files, wherein each configuration file defines different relationships between purchasing parameters and one or more tag names.

40. (Previously presented) The device of claim 31, wherein the markup language is eXtensible Markup Language (XML).

41. (Currently amended) A computer accessible medium, comprising program instruction ~~configured to~~ that when executed by a computer implement:

receiving one or more input documents in a markup language, wherein the input documents comprise a purchasing request from a supplier, wherein each input document comprises one or more tag names specifying purchasing parameters for the purchasing request;

parsing each of the one or more input document to identify each of the one or more tag names;

generating one or more data objects each corresponding to a respective tag name;

identifying one or more purchasing parameters related to one or more of the identified tag names; and

generating one or more output documents in the markup language specifying a purchasing order to the supplier corresponding to the purchasing request specified in the one or more input documents, wherein the one or more output documents comprise:

one or more tag names corresponding to the one or more identified purchasing parameters, and

one or more data attributes each corresponding to one of the one or more tag names corresponding to the one or more identified purchasing parameters.

42. (Currently amended) The computer accessible medium of claim 41, wherein the program instructions, when executed, ~~are further configured to~~ implement:

generating a data object tree of the one or more data objects based on relationships between the corresponding tag names as defined in the one or more input documents; and

wherein said identifying one or more purchasing parameters comprises traversing the data object tree comparing each data object of the tree to one or more data objects in a database specifying the one or more purchasing parameters.

43. (Currently amended) The computer accessible medium of claim 42, wherein in said identifying one or more purchasing parameters the program instructions ~~are~~ further ~~configured to~~ implement identifying the one or more purchasing parameters based upon a configuration file defining relationships between the purchasing parameters and the one or more tag names.

44. (Previously presented) The computer accessible medium of claim 43, wherein the configuration file is one of a plurality of configuration files, wherein each configuration file defines different relationships between the purchasing parameters and the one or more tag names.

45. (Previously presented) The computer accessible medium of claim 41, wherein the markup language is eXtensible Markup Language (XML).